A Methodological View on Robustness Testing of Rule-Based Knowledge Systems

Joachim Baumeister, Jürgen Bregenzer, Frank Puppe
Department of Computer Science
University of Würzburg, 97074 Würzburg, Germany
phone: +49 931 8886740 fax: +49 931 8886732
{baumeister, bregenzer, puppe}@informatik.uni-wuerzburg.de

Abstract: Besides the evaluation of intelligent systems with respect to the accuracy the robustness of such systems is often an important issue to consider. Robustness relates the correct output of a system to possibly incorrect input or possibly biased knowledge included in the system. In this paper we extend the approach by Groot et al. [GvHtT00] by a pre-analysis step considering the used case base and knowledge base, respectively. Appropriate measures are introduced that determine the suitability of robustness tests for a given case/knowledge base. Furthermore, we motivate that the application of background knowledge is necessary in order to obtain reliable results from robustness studies.

1 Introduction

The validation and verification of intelligent systems has been investigated thoroughly over the last 20 years in knowledge engineering research, e.g. [AL91, PSB92, Kna00]. Commonly, the accuracy of such systems is of prime importance, i.e. the sound and complete derivation of solutions for a given input. However, besides these standard measures the *robustness* of the system is also an interesting characteristic to be considered when evaluating the usefulness of an intelligent system. The robustness considers the correct output of the system w.r.t. incorrect input values and possibly biased knowledge in the system. Thus, robustness testing investigates the correct/expected behavior of the knowledge system even if some input values were falsely entered (e.g. due to typos of the user), or when some rules included in the knowledge base were wrongly formalized (e.g. due to initial errors of the developer). Similarly, in software engineering the robustness is also tested by malicious user entries. Initially, testing the robustness of rule-based systems was described by Groot et al. [GvHtT00] with the application of *torture tests*. In general, torture tests are an extension of the well-known empirical testing method. With empirical testing a collection of previously solved and correct test cases is given to the knowledge system as input. Then, for each test case the solutions derived by the knowledge system are compared with the solutions already stored in the cases. Typically, measures like the precision, the recall or the E-measure are used for a quantitative comparison of the two solution sets. Torture tests run a series of empirical tests by degrading the quality of settings, e.g. by gradually degrading the input quality or by gradually worsening the quality of the used knowledge

base. Within such a degradation study the changing performance of the knowledge system is measured.

In this paper, we introduce a methodological approach for implementing degradation studies and we extend the possible torture operations introduced by Groot et al. In Section 2 we describe the basic measures for evaluating and comparing the robustness of knowledge systems. We introduce the concept of degradation studies in Section 3 and we emphasize the importance of a pre-analysis of the rule base and the case base used for the tests. We discuss that degradation studies should not be considered as black-box tests and that sometimes background knowledge is essential. In Section 4 we summarize the presented approach and will point to future work.

2 Evaluating the Robustness of Knowledge Systems

The robustness of a knowledge system considers the quality of the output depending on changing quality of the input. For a systematic description we introduce some basic notions.

Basic Definitions We distinguish *input values* given to a knowledge system and *output values* that are derived by the knowledge system for a given set of inputs. More formally, we define Ω_{obs} to be the (universal) set of observable *input values* $f = a : v$, where $a \in \Omega_a$ is an attribute and $v \in dom(a)$ is an assignable value. Let Ω_{sol} be the universe set of (boolean) *output values*, i.e. solutions derivable by the knowledge system. A test-case c is defined as a tuple $c = (\, OBS_c, SOL_c \,)$, where $OBS_c \subseteq \Omega_{obs}$ is the *problem description* of the case, i.e. the observed inputs of the case c; $OBS_c = \{f_{1,c}, \ldots, f_{n,c}\}$. The set $SOL_c \subseteq \Omega_{sol}$ contains the (correct) solutions of case c. In general, we define a quality function for a given knowledge system and a collection of test cases as follows.

Quality Function Let C be the universe of test cases, i.e. containing all possible combinations of input values and let K be the knowledge base. Then, $q : C \times K \rightarrow [0, 1]$ is a *quality function* comparing the expected solution documented in a case c and the solution of c derived by the system K. Examples for a quality function are the *precision*, the *recall* or applications of the *E-measure*.

When the input quality is degraded, then the system should show a monotonically degrading output quality, i.e. no fluctuating output quality. In consequence, the output quality of the system is more predictable. More formally we define the monotonic derivation quality of a knowledge system as follows.

Monotonic Derivation Quality For a knowledge system K let $C = (c_1, \ldots, c_n)$ be a sequence of cases sorted according to their input quality in ascending order; further let q be a quality function. Then the knowledge system shows a *monotonic derivation quality*, if $q(c_i, K) \leq q(c_{i+1}, K)$ for all $1 \leq i \leq n$.

This criterion is a necessary requirement for any knowledge system, that should be considered to be *robust*. Groot et al. [GtTvH05] introduce measures for comparing two knowledge systems w.r.t. their robustness: The quality value and the rate of quality change.

Quality Value For two knowledge systems K_1 and K_2 we say that K_1 is more robust than K_2 for a given quality function q, if for any test case c the output quality of K_1 is higher than the output quality of K_2, i.e. $q(c, K_1) > q(c, K_2)$.

Whilst the *quality value* considers the isolated behavior of the knowledge system for a given test case, the *rate of quality change* emphasizes the comparison of dynamic behavior of the knowledge systems.

Rate of Quality Change For two knowledge systems K_1 and K_2 and a quality function q, we say that K_1 is more robust than K_2, if for any qualitatively ordered sequence of test cases the average quality of the output of K_1 decreases more slowly than for K_2.

The measures presented so far are useful for comparing different aspects of the robustness of two knowledge bases. They can be intuitively applied for an analysis of a degradation study that is explained in the next section. However, in this paper we will motivate that a thorough degradation study also should carry out a detailed analysis of the case base and knowledge base, respectively.

3 Degradation Studies

For the degradation studies a sufficient case base is used that should contain a representative selection of test cases. We describe measures that are applied in a preliminary analysis of the used test case base. Furthermore, for a reasonable evaluation of the knowledge with robustness tests the characteristics of the knowledge base should be also considered. We will argue that the knowledge base should not be used as a black box since the results of a degradation study heavily depend of the structure and semantics of the knowledge base. Then we present a description of the degradation studies that will consist of iterative applications of torture tests.

3.1 Pre-Analysis: Case and Knowledge Base Properties

The properties of the used case base and knowledge base should be analyzed before starting any degradation study. It is easy to see that an insufficient characteristic of the cases will not yield sound insights w.r.t. the robustness of the knowledge base. At best, the cases are equally distributed over the collection of possible findings and number of diagnoses. Furthermore, we present measures investigating the characteristics of the knowledge base. Some structures contained in a knowledge base can influence the results of torture tests.

Average Number of Cases (NOC) In a first step, the used cases can be characterized by the *average number of cases* for each solution, i.e. the expectation value with standard deviation of cases is computed for each solution. A low number of cases makes it difficult to generalize the obtained results of the torture tests since only a small spectrum of the real world is possibly covered. A high deviation may indicate that some solutions only contain very few cases or a high number of cases. Consequently, the robustness can be very low or high for some diagnoses.

Average Number of Findings (NOF) A second analysis should consider the *average number of findings* contained in the cases, i.e. the expectation value with standard deviation of findings for the cases is computed. A low expectation value of findings can imply that the case base is not suitable for an extensive degradation study, because even smaller modifications of the cases may imply a large percentage of noise within the input data.

Average Number of Rules (NOR) For degradation studies concerning the modification of the rule base the *average number of rules* for each solution, i.e. the expectation value with standard deviation of derivation rules for each solution, is also an interesting measure. A low number of average rules or high deviation values can indicate that the results of the robustness studies may not be representative for all solutions contained in the knowledge base. The coverage of test cases has been investigated more thoroughly e.g. by [Bar99]. The analysis of the test cases is important for evaluating the sound execution of the degradation studies.

Types of Rules In classical systems the knowledge base only contained rules directly deriving a solution for a given condition. However, for real-world applications the rule base can contain more refined types of rules. We distinguish three basic types of rules (cf. [Bau04] for a more formal description):

1. *Diagnostic rules*: For a given condition the rule derives a specified solution. In detail, we distinguish diagnostic rules that derive a solution categorically and diagnostic rules deriving a solution using evidential categories, e.g., scores or probabilities.
2. *Abstraction rules*: For a given condition such rules derive the value for an intermediate abstraction, that in turn can be also used in further rule conditions. Abstraction rules are suitable for improving the reuse of existing knowledge or to enhance the design/understandability of a knowledge base.
3. *Indication rules*: Such rules are used to implement an adaptive and efficient dialog of the system with a user. For a given rule condition the rule indicate new questions/inputs to be presented to the user.

We see that a rule base containing not only diagnostic rules but also abstraction and indication rules is much more difficult to test for robustness. Thus, eliminating a specific question can prevent the system to ask the original questions and thus entirely change the semantics of the case. For example, decision trees are a prominent representation often implemented using indication rules. The availability of abstraction rules introduces rule chains in the knowledge base and therefore the elimination or modification of a specific

input value can also change large parts of the original case. For this reason, no accurate evaluation may be possible.

Complexity of Rules For evaluating the results of degradation studies the complexity of the included rules is also an interesting measure. In general, the complexity of a rule is calculated by the number of simple conditions (i.e., evaluating the value of a single input) included in the rule condition. This measure can be integrated in the previously described measure *Average Number of Rules (NOR)* by weighting the single rules with their complexity.

3.2 Degradation by Torture Tests

In the following we describe the implementation of degradation studies. In general, a degradation study consists of a sequence of torture tests with decreasing input quality. With a degradation study one observes the changing behavior of a knowledge system when the input quality is gradually worsened. We distinguish the following types:

1. **Torture by change**
   (a) **Change input values**: An increasing number of values contained in the cases is modified. With this type of torture test the robustness of the knowledge system w.r.t. an incorrect data acquisition can be evaluated.
   (b) **Change knowledge elements**: An increasing number of rules contained in the knowledge base is modified. Then, for example rule actions are slightly changed. With this test the robustness of the knowledge system w.r.t. some biased or faulty knowledge acquisition can be evaluated. E.g., how dramatically do some incorrect rules worsen the accuracy of the system?
2. **Torture by deletion**
   (a) **Delete input values**: An increasing number of randomly selected values contained in the cases are not given as an input to the knowledge system. The test is suitable to evaluate the robustness of the system w.r.t. missing values or incomplete data acquisition.
   (b) **Delete knowledge elements**: An increasing number of randomly selected rules contained in the knowledge base is not used for the derivation of the solutions. The test can be useful to evaluate the robustness of the knowledge base w.r.t. to an incomplete knowledge acquisition.

We see that there exist four different types of torture tests to be used in separate degradation studies. In the following section we discuss some background knowledge that can be used within the particular torture tests.

3.3 Inclusion of Background Knowledge

For a reliable degradation study we have to consider the measures introduced in Section 3.1: For example, we expect no reliable results for a rule base containing an extensive amount of indication and abstraction rules, since even small modifications or eliminations of input values can basically change the tortured case. In contrast, we expect the system to have a good robustness, if only diagnostic rules are included with a high share of simple conditions. A more realistic degradation study can be conducted if additional background knowledge is included. We present two useful types of background knowledge:

Ambivalence of input values For a real world application the possible value range of some input values is more ambivalent than for other input values, i.e., for some input values the user will more likely mix up the values and enter a wrong answer. In a degradation study such ambivalent input values should be more likely tortured than others, for which the user is unlikely to be confused about the correct answer. If the developer of the rule base marks input values as *possibly ambivalent*, for which he/she thinks that the possible values cannot be easily discriminated, then the torture test can use this background knowledge during the selection of input values to be modified/removed. For the *torture by change*-test (1a) possibly available similarity knowledge of the input values can be used to select a similar value.

Importance of input values In contrast to the ambivalence of input values the developer can mark some input values to be important for the reasoning task. In consequence, *important input values* are categorically excluded from the elimination/modification torture tests. For example, such important input values may be contained in the condition of indication or abstraction rules, and will ensure that the basic logic of the particular case is not changed. However, when marking important input one has to consider that these values are never changed throughout the torture tests. In consequence, values should be only marked as *important* if they are not already identified as an *ambivalent input*.

3.4 Discussion

The previous insights imply that robustness tests cannot be considered as a black-box testing method, but a thorough analysis of the case base and rule base, respectively, should be carried out before the actual implementation. Thus, the appropriateness of the case base and the rule base need to be investigated before starting the degradation studies. In a first case study we re-implemented the degradation studies described in [GtTvH05] and performed experiments with the same plant knowledge base as mentioned in the paper. We could reproduce the derived precision/recall values in our own degradation studies. This plant rule base was formalized using the Diagnostic Score pattern [Pup00], which proposes to formalize the knowledge by rules with single conditions. Thus, the basic idea of the Diagnostic Score pattern is to rate all possible individual input values w.r.t. given solutions (diagnoses). For each solution we list all input values confirming or disconfirming the solution and rate their confirmation strengths using symbolic categories. Typically, for

each input value–solution relation only a small confirmation category is defined. As expected the robustness of the system was very good, since the distributions of the rule base and case base were quite acceptable (expectation values with standard deviation in parentheses): NOR=73.7 (8.5), NOF=26.8 (4.1), NOC=2.0 (1.2), no indication and abstraction rules. However, we failed to transfer the procedure to a larger rule base containing 50% diagnostic rules, 25% indication rules and 25% abstraction rules (total of 8475 rules). Since, often important input values were eliminated/changed the semantics of the particular case was completely changed and no solution could be derived at all. An open problem is the question of how to rate such cases: Since no solution was derived the test would compute a precision equal to zero, although the system performed correct for the given input, i.e. due to the changed semantics of the original case no appropriate solution could be found. This led us to the conclusion that the implementation of a pre-analysis of the knowledge base is very important and that results have to be scrutinized w.r.t. the properties of the current application domain. An important issue is the question when to abort the degradation study, e.g. when a pointless low input quality value is reached. We propose that for example for the torture tests considering the elimination/modification of input values we should not consider a quality range from 100% to 0% but an even smaller interval. If ambivalent input values are defined, then the degradation study should focus on these input values but not on the entire collection of input values. Additionally, the acquisition of important input values is also very important for preserving the original semantics of the cases. We are currently planning to implement the considerations discussed above in order to perform a knowledge-intensive degradation study with a series of knowledge bases.

4 Conclusion and Outlook

In this paper we presented a methodological view of degradation studies. Such studies are carried out for testing the robustness of rule-based knowledge systems. In comparison to the original introduction of degradation studies by [GvHtT00] we added a pre-analysis of the knowledge base and the used cases, respectively, and motivated that robustness testing should not be considered as a black-box testing method. We discussed particular background knowledge that need to be added for complex rule bases. Furthermore, the application of background knowledge can help to obtain more reliable results of the degradation studies. We also motivated that degradation studies heavily depend on the underlying formalization pattern of the rule base: Whereas rule bases implementing a Diagnostic Score pattern are well-suited for torture tests we see that systems built by decision trees are not as robust.

In the future we are planning to empirically evaluate the discussed considerations. First experiments show promising results but a more detailed analysis will uncover the variable importance of the particular measures and types of background knowledge. A more detailed analysis will consider the differences between knowledge bases that were build using different formalization patterns, e.g. Diagnostic Scores vs. decision trees. We expect knowledge systems formalized by Diagnostic Scores to be more robust than systems that were implemented by the latter. Additionally, the use of threshold values for the number

of modified (tortured) elements is necessary for yielding reliable results of the degradation study, since we cannot expect a correct behavior of the system in a totally altered environment. For example, even for a defined collection of ambivalent values only a limited percentage of input values should be tortured within a study.

References

[AL91] Marc Ayel and Jean-Pierre Laurent. *Validation, Verification and Test of Knowledge-Based Systems*. Wiley, 1991.

[Bar99] Valerie Barr. Applications of Rule-Base Coverage Measures to Expert System Evaluation. *Knowledge-Based Systems*, 12:27–35, 1999.

[Bau04] Joachim Baumeister. *Agile Development of Diagnostic Knowledge Systems*. AKA Verlag, DISKI 284, 2004.

[GtTvH05] Perry Groot, Annette ten Teije, and Frank van Harmelen. A Quantitative Analysis of the Robustness of Knowledge-Based Systems through Degradation Studies. *Knowledge and Information Systems*, 7:224–245, 2005.

[GvHtT00] Perry Groot, Frank van Harmelen, and Annette ten Teije. Torture Tests: A Quantitative Analysis for the Robustness of Knowledge-Based Systems. In *Knowledge Acquisition, Modeling and Management*, LNAI 1319, pages 403–418, Berlin, 2000. Springer Verlag.

[Kna00] Rainer Knauf. *Validating Rule-Based Systems: A Complete Methodology*. Shaker, Aachen, Germany, 2000.

[PSB92] Alun Preece, Rajjan Shinghal, and Aida Batarekh. Principles and Practice in Verifying Rule-Based Systems. *The Knowledge Engineering Review*, 7 (2):115–141, 1992.

[Pup00] Frank Puppe. Knowledge Formalization Patterns. In *Proceedings of PKAW 2000*, Sydney, Australia, 2000.